ROB313 A2

Ethan Rajah (1006722370)

March 2 2023

1 Objectives

The primary objective of this assignment is to implement more advanced generalized linear models for greater accuracy when predicting on data. Specifically, Radial Basis Functions (RBFs) and greedy regression algorithms are explored within questions 3 and 4. The process of determining optimal hyper parameters, θ and λ for the RBF model is explored within question 3, which allows for an analysis of the effects of increasing and decreasing both parameters. Furthermore, an alternate approach is taken with the greedy regression algorithm, where basis functions are created and chosen to minimize residual error between the actual data and the predicted results. The basis functions that minimize this error are coupled with their respective weights to create a sparse regression model for the data. This assignment allows one to analyze the benefits and losses of using each of the model approaches, particularly through run time and test root mean square error differences. Prior to this analysis, the assignment provides an opportunity to become familiarized with the process of optimizing regression models such as the kernel form of the generalized linear model by minimizing the least squares loss function using different regularization forms.

2 Code Structure and Strategies

2.1 Question 1

The first question involves deriving a closed form expression for the weights of the generalized linear model using the least squares loss function and the Tikhonov regularization equation. This involved converting the given equations into their vectorized forms and differentiating with respect to the weight vector to minimize the least squares loss. Then, the weight vector can be isolated for to give the final form of the expression. For reference, Γ is a positive semi-definite matrix that is defined within the formulation for the Tikhonov regularization equation.

2.2 Question 2

The second question involves deriving an expression for $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_N\}^T$ using the kernelized form of the derived GLM, $\hat{f}(\mathbf{x},\mathbf{w}) = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}, \mathbf{x}_i)$. This is similar to the derivation for the weights using the dual representation form seen in lecture. The objective function, consisting of the

least squares loss function and a regularization term, is used with the GLM to vectorize the system and and differentiate with respect to α. The regularization term, $\lambda \sum_{i=1}^{N} \alpha_i^2$ determines how well the regression model fits to the data. This is useful for question 3, where varying the regularization term, λ can be seen in the root mean square error of the test data.

2.3 Question 3

As mentioned in the objectives section of the document, question 3 aimed to provide insight on the process of using RBF models to predict data, as well as how to select the hyper parameters, θ and λ, so that the regression model is neither overfit nor underfit. This is useful for predicting on new data as the model becomes better at dealing with variance and noise in future measurements. If the regression model is underfit, it will miss the characteristic behaviour of the data, however, if it is overfit, it will not be able to account for changes in the behaviour of the model when predicting on data that it has not been trained on. This is the function of varying the value of lambda. Furthermore, varying θ alters the variance of the kernel calculations. More specifically, it alters the width of the basis functions, so that when the variance/width is greater, there is a greater measure of similarity between the vectors compared in the kernel, depending on the location of the data with respect to the landmark/basis function.

The RBF model with the optimized hyper parameters were determined using a Gaussian kernel, as defined in lecture, as well as using Cholesky factorization to solve for the vector, α. Two loops were used to iterate through the possible values of theta and lambda. The Gaussian kernel was first used to create the gram matrix, K, using the training data and the current value of theta. Then, since $\alpha = (K + \lambda I)^{-1}\mathbf{y}$, as derived in lecture, and is semi positive definite, the Cholesky factorization of $(K + \lambda I)$ was computed and used to solve the linear equation, $R^T R\alpha = y$ for α. The RBF model was then formed by taking the dot product of the Gaussian kernel (comparing the validation and training sets) and α. In this case, the training data is used as the landmarks (basis vectors) for predicting on the validation set. The prediction result of the RBF model was compared against the exact data to calculate the root mean square loss of the model with the specified values of θ and λ. This was done for every possible combination of θ and λ. The parameters that resulted in the lowest validation RMSE were chosen as the optimal hyper parameters for the regression model.

To assess the performance of the RBF model with the optimal hyper parameters, predictions were done on the test set. Both the training and validation sets were used together to complete this assessment. Specifically, the gram matrix corresponding to the chosen value of θ and α vector derived from the chosen value of λ were used to form the final RBF model, where the Gaussian kernel measured the similarity between the test and training sets. Both the Mauna Loa and Rosenbrock datasets were applied to the RBF algorithm and the test RMSE values were calculated from the resulting prediction.

2.4 Question 4

This question focused on using a dictionary of basis functions to create a greedy algorithm that aims to minimize the residual loss of the training data. The orthogonal matching pursuit algorithm, a forward greedy algorithm for regression, was implemented as the method of selecting basis functions from a dictionary of 200 functions. To create the dictionary, the behaviour of the Mauna Loa dataset was observed graphically. One can notice that the behaviour of this data consists of some polynomial factors with sine and cosine terms producing oscillatory behaviour. Based on

this observation, 200 basis functions were created by generating polynomial terms up to degree 7, which was arbitrarily chosen, and sine/cosine terms using variations of frequency, ω to fill the rest of the dictionary. More specifically, the basis functions that needed to be created after adding the polynomial terms were split in size, where half were made into sine functions and the other half cosine functions, both with varying frequency.

With the dictionary complete, a 'candidates' array was created to store the indices of the candidate basis functions and a 'selected' array was created to store the indices of the basis functions we've chosen so far. Now within the algorithm, the orthogonal matching pursuit metric, $J(\phi_i) = \frac{(\boldsymbol{\Phi}(:,i)^T \boldsymbol{r}_k)^2}{\boldsymbol{\Phi}(:,i)^T \boldsymbol{\Phi}(:,i)}$ was calculated over all the candidate basis functions. Choosing the maximum element from the generated matrix is analogous to choosing the basis function that provides the greatest residual decrease. Moreover, the index corresponding to the basis function that gives the greatest residual decrease was added to the selected array, and removed from the candidates array so that it would not be chosen in future iterations of the algorithm. Using the current selected basis functions, the weights of the regression model were calculated. Since there was no information on if the matrix generated by the current basis vectors within the selected array, Φ_k, was semi-positive definite, SVD was used to solve the linear equation, $\Phi_k \boldsymbol{w}_k \approx \boldsymbol{y}$, as it is a more robust solving approach. Then, using the expression for the weights derived in class in terms of the SVD matrices, $\hat{w} = U_1 S_1^{-1} V^T \boldsymbol{y}$, the weights were calculated and used to generate the current regression model and a prediction set for the training data. For this question, the training data consisted of both the training and validation sets together.

The forward greedy algorithm runs within a while loop that continues until there are no more candidates left, or if the error is within a small enough bound, ϵ. For this question, it is asked to use the minimum description length (MDL) as stopping criterion for the greedy algorithm. It is also noted that this metric will decrease as the model complexity grows, and increase as over fitting begins to occur. For this reason, after creating the prediction set using the current model which encapsulates the selected basis functions at the k-th iteration of the algorithm, the MDL was calculated by determining the residual loss between the prediction set and true data. This was then used to determine the l_2 loss and thus, the MDL for the k-th iteration. The code was structured so that if the MDL calculation was greater than the previous, the algorithm would terminate as it would indicate that the model was becoming over fit. After the algorithm was broken out of, the current selected basis functions and respective weights were used to represent the final model. This model was applied to the test set and plotted to observe the accuracy when it faced new data.

I also plotted the fit of the model for each iteration of the algorithm to track its growth in accuracy with a greater number of iterations. Some of these plots can be viewed within the **Appendix**.

3 Results and Discussion

3.1 Question 1 Derivation

Using a generalized linear model, $\hat{f}(\mathbf{x},\mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$, a least squares loss function, and the general Tikhonov regularization, a closed form expression for the weights can be derived. The derivation begins with the Tikhonov regularization equation, which can be vectorized to remove the sums within the loss function. The vector form of the generalized linear model is used to vectorize

the Tikhonov equation:

$$L = (\sum_{i=1}^{N}(y_i - w_0 - \sum_{j=1}^{M-1} w_j\phi_j(\mathbf{x}_i))^2 + \sum_{i=1}^{M}\sum_{j=1}^{M}\Gamma_{ij}w_{i-1}w_{j-1})$$
$$L = (\sum_{i=1}^{N}(y_i - \phi(x_i)\mathbf{w})^2 + \mathbf{w}^T\Gamma\mathbf{w})$$
$$L = (\mathbf{y} - \Phi\mathbf{w})^T(\mathbf{y} - \Phi\mathbf{w}) + \mathbf{w}^T\Gamma\mathbf{w}$$

With the simpler vectorized form of Tikhonov's regularization, we can differentiate the loss function to minimize with respect to the weights, $\mathbf{w}$. After minimizing, the weight vector can be isolated for to obtain its closed form expression. This is shown below:

$$\begin{aligned}
\frac{\partial L}{\partial w} &= 2(\mathbf{y} - \Phi\mathbf{w})^T(-\Phi) + 2\Gamma\mathbf{w} \\
0 &= -2(\mathbf{y} - \Phi\mathbf{w})^T\Phi + 2\Gamma\mathbf{w} \\
0 &= -2\Phi^T(\mathbf{y} - \Phi\mathbf{w}) + 2\Gamma\mathbf{w} \\
0 &= -\Phi^T\mathbf{y} + \Phi^T\Phi\mathbf{w} + \Gamma\mathbf{w} \\
\Phi^T\mathbf{y} &= (\Phi^T\Phi + \Gamma)\mathbf{w} \\
\mathbf{w} &= (\Phi^T\Phi + \Gamma)^{-1}\Phi^T\mathbf{y}
\end{aligned}$$

3.2 Question 2 Derivation

Using the generalized linear model, $\hat{f}(\mathbf{x}, \boldsymbol{\alpha}) = \sum_{i=1}^{N}\alpha_i k(\mathbf{x}, \mathbf{x}_i)$ we can derive a computational strategy to estimate $\boldsymbol{\alpha}$. This can be done by minimizing the least squares loss function that includes the regularization terms (the objective function) and solving for an expression for $\boldsymbol{\alpha}$. The simplification of this loss function using the vectorized form of the generalized linear model is shown below, where K is the symmetric Gram matrix consisting of kernel vectors.

$$L = \sum_{i=1}^{N}(y_i - \hat{f}(\mathbf{x}, \boldsymbol{\alpha}))^2 + \lambda\sum_{i=1}^{N}\boldsymbol{\alpha}_i^2$$
$$L = \sum_{i=1}^{N}(y_i - \sum_{i=1}^{N}\alpha_i k(\mathbf{x}, \mathbf{x}_i))^2 + \lambda\sum_{i=1}^{N}\boldsymbol{\alpha}_i^2$$
$$L = \sum_{i=1}^{N}(y_i - K\boldsymbol{\alpha})^2 + \lambda\boldsymbol{\alpha}^T I\boldsymbol{\alpha}$$
$$L = (\mathbf{y} - K\boldsymbol{\alpha})^T(\mathbf{y} - K\boldsymbol{\alpha}) + \lambda\boldsymbol{\alpha}^T I\boldsymbol{\alpha}$$

Now that we have a more compact form of the loss function with regularization, we can differentiate it with respect to $\boldsymbol{\alpha}$ to obtain an expression for the weights that minimizes the objective

function:

$$\begin{aligned}
\frac{\partial L}{\partial \alpha} &= 2(\mathbf{y} - K\boldsymbol{\alpha})^T(-K) + 2\lambda\boldsymbol{\alpha} \\
0 &= -K^T(\mathbf{y} - K\boldsymbol{\alpha}) + \lambda\boldsymbol{\alpha} \\
0 &= -K^T\mathbf{y} + K^TK\boldsymbol{\alpha} + \lambda\boldsymbol{\alpha} \\
K^T\mathbf{y} &= (K^TK + \lambda I)\boldsymbol{\alpha} \\
K\mathbf{y} &= (KK + \lambda I)\boldsymbol{\alpha} \\
\boldsymbol{\alpha} &= (KK + \lambda I)^{-1}K\mathbf{y}
\end{aligned}$$

This result is not the same as the expression we derived in class using the dual representation because in class, we used a generalized linear model of the form seen in Question 1. However, for this problem, we use a general model that is constructed using the kernel vectors, rather than the basis feature vector represented as $\phi(x)$. Kernel and feature vectors were shown in lecture to be related through the expression, $\phi(x_i)^T\phi(x) = k(x_i, x)$. Moreover, we also fit to the weight vector $\boldsymbol{\alpha}$, rather than $\mathbf{w}$ which are related by the form, $\mathbf{w} = \Phi^T\boldsymbol{\alpha}$. These differences result in a slight variation in the resulting expression for the weight vector, $\boldsymbol{\alpha}$, in comparison to the result seen in lecture.

3.3 Question 3

Results	Test RMSE	Theta	Lambda
Mauna Loa	0.149	1.0	0.001
Rosenbrock	0.148	2.0	0.001

Figure 1: Test RMSE results for RBF regression

The test RMSE results for both the Mauna Loa and Rosenbrock datasets can be seen in **Figure 1**. In generating the RBF models for each dataset, it was found that a regularization factor of $\lambda = 0.001$ provided the best fit with respect to generating models that were neither over fit, nor under fit. However, both datasets varied in their optimal value of θ as Mauna Loa had a $\theta = 1.0$, whereas the Rosenbrock dataset had a $\theta = 2.0$. As previously mentioned, larger values of θ mean a wider basis function, so in the case of using the Gaussian RBF kernel, there is more variance when θ increases, therefore allowing for more similarity (greater region of influence) between the basis vectors and points that are further away. Moreover, it was found that a larger value of θ than that used for modelling the Mauna Loa dataset provided the greatest reduction in loss for the RBF model. As seen in **Figure 1**, both test RMSE values are close to 0.15. This shows that although this model is an improvement to the results seen using the k-NN algorithm or other generalized linear model for regression in assignment 1, there is still some room for improvement to the model accuracy. Furthermore, the RBF regression is $O(n^2)$ in space complexity and is $O(n^3)$ in time complexity when solving for α, which is not desirable when dealing with larger data sets such as mnist. For this reason, greedy algorithms, which only use a subset of basis functions to generate a sparse RBF regression model, tend to be used as they can reduce space and time complexity. An example of this is seen through question 4.

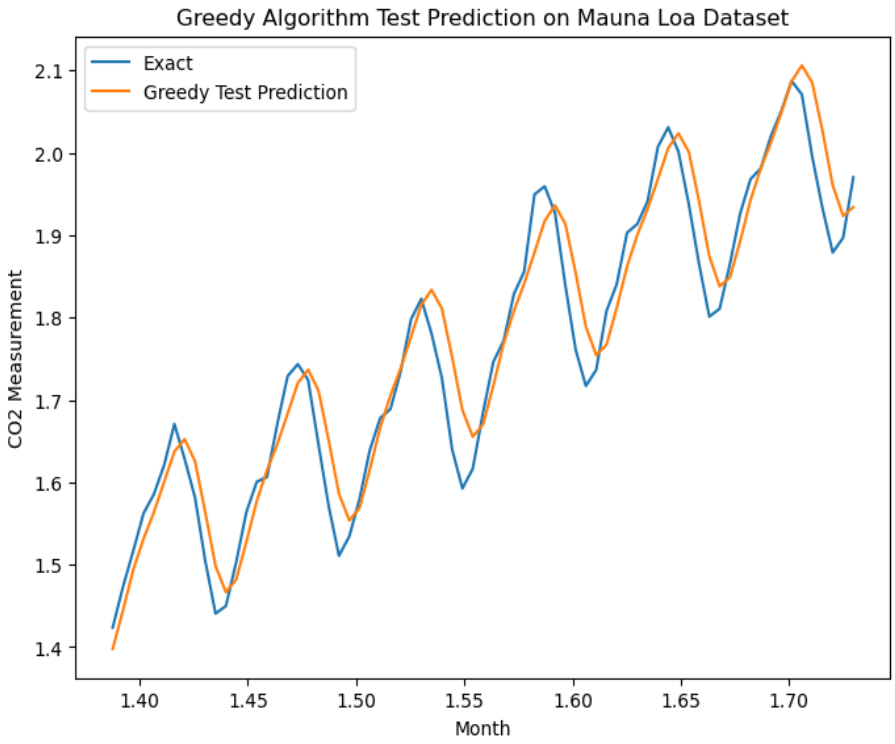


Figure 2: Test prediction for Mauna Loa data using sparse model generated in the greedy algorithm

3.4 Question 4

As seen in **Figure 2**, the results from the orthogonal matching pursuit metric and algorithm provides an accurate model of the Mauna Loa data, even when introduced to new data. This figure shows that the model was able to successfully predict the behaviour of the test data, specifically with a test RMSE of 0.046. This is a significant improvement to the results seen for the Mauna Loa data set in question 3 using the RBF regression model. This is likely because within this greedy algorithm, we have a metric for choosing a subset of basis functions that contribute to the greatest residual decrease, while not over fitting the model. Furthermore, the use of the MDL metric aids in ensuring that the model does not become over fit, which can be seen in **Figure 2** as it closely matches the behaviour (within an error range) of the test data, without having previously seen the data. For comparison, we saw with the effects of having an overfit model when implementing the k-NN algorithm in assignment 1. Even though the validation accuracy of the model was high, the test prediction for Mauna Loa was a horizontal line, rather than a function resembling sinusoidal behaviour like the true data.

When testing the results of this algorithm, I found that although I was creating sine and cosine basis functions using varying frequencies, the final model would only exhibit small oscillatory behaviour. For this reason, I decided to multiply each frequency used by π to amplify the oscillations, which provided much better results for the final model. Moreover, I found that by increasing the maximum degree of the polynomial basis functions generated, the test RMSE would increase. This is likely due to the dominant presence of sinusoidal behaviour within the Mauna Loa data, so I

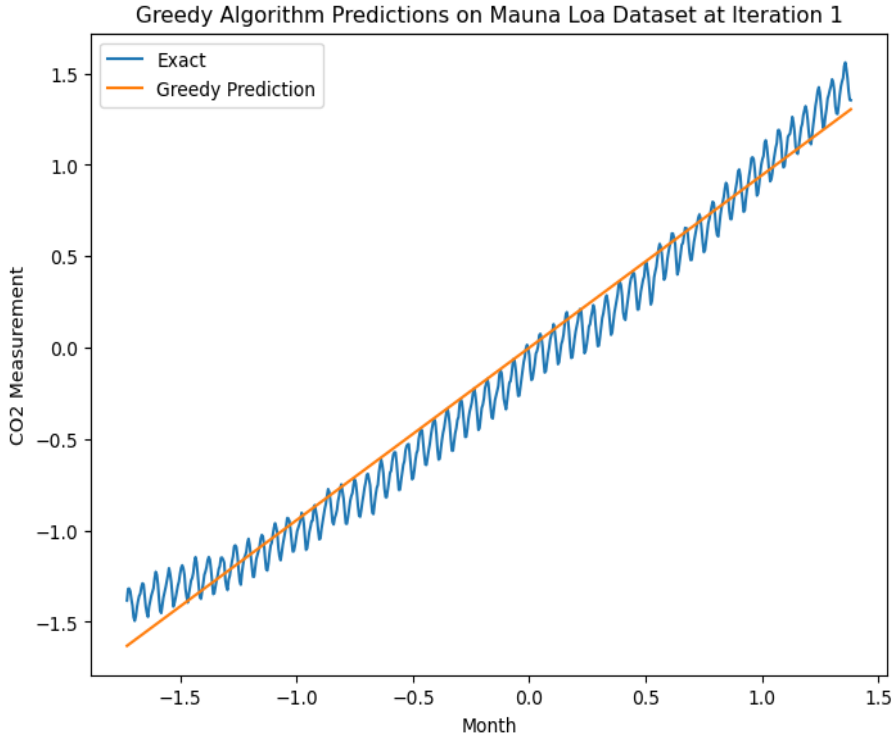
decided to reduce my maximum polynomial degree to 3 as it provided the best test RMSE results. This makes sense for the Mauna Loa data because when observing the true data, it seems that a model with polynomial terms of no more than a degree of 3 is sufficient for modeling its growth behaviour.

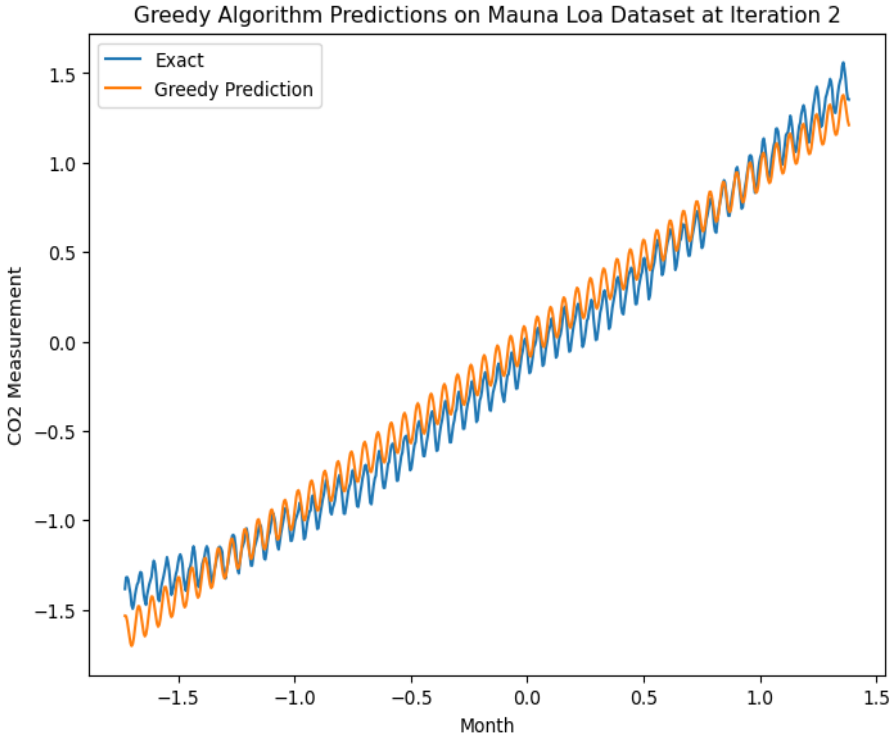
As mentioned towards the end of the analysis for question 3, the sparse regression model generated from the greedy algorithm aids in reducing space and time complexity of the program. The final model was generated using 14 basis functions from the 200 available functions in the dictionary as the algorithm terminated after the k=14 iteration. As mentioned in lecture, sparsity also aids in preventing over fitting as it is a form of regularization for a model. As such, it leads to a simpler model because it learns what model parameters can be dropped, thus leading to a simpler, more interpretable model.

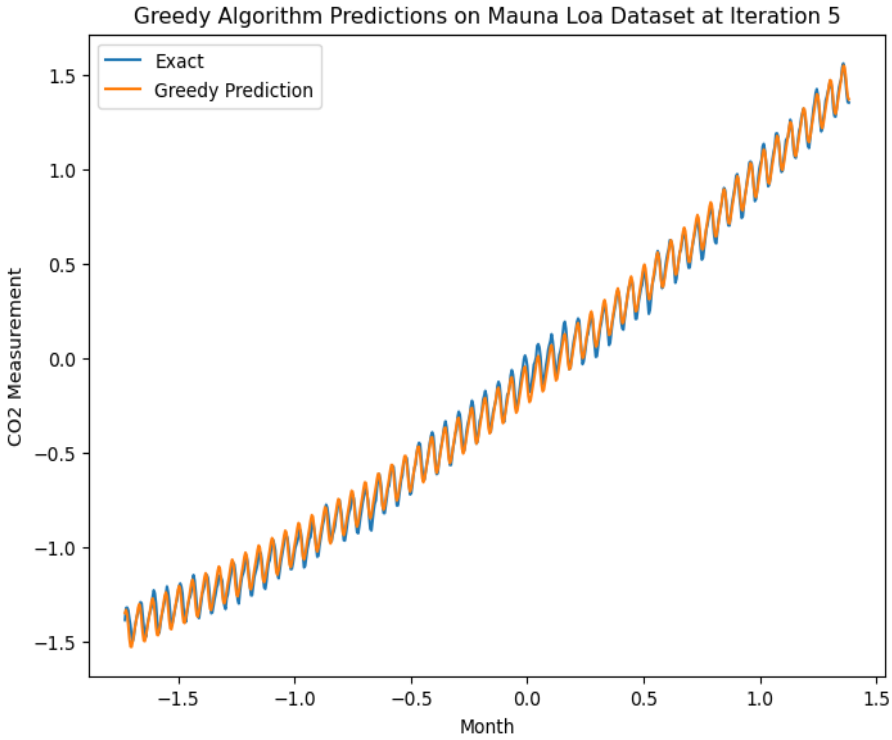
4 Appendix

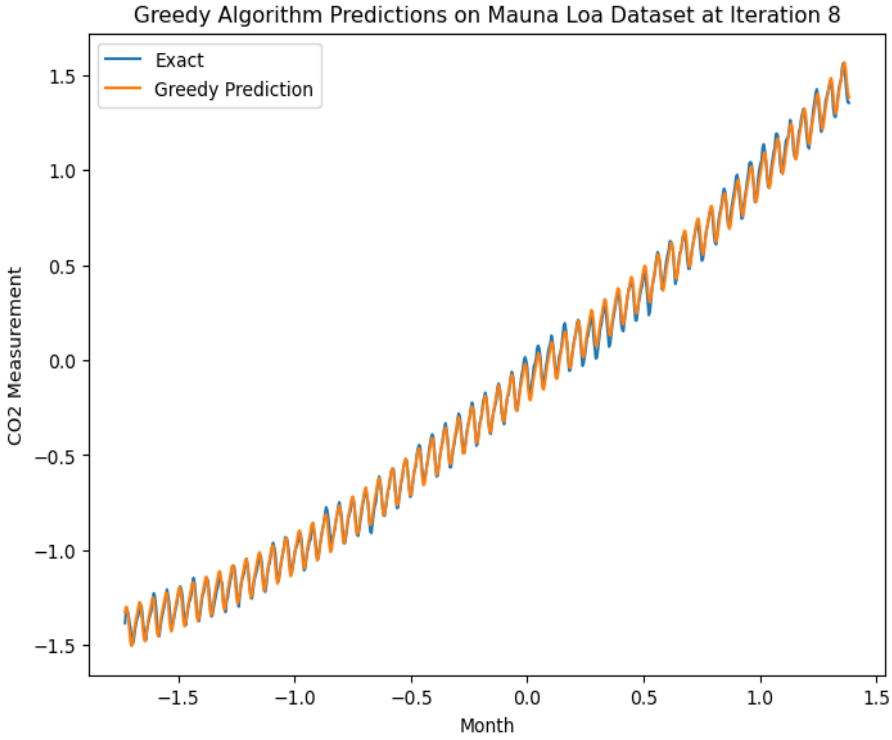
5 Greedy Algorithm Predictions

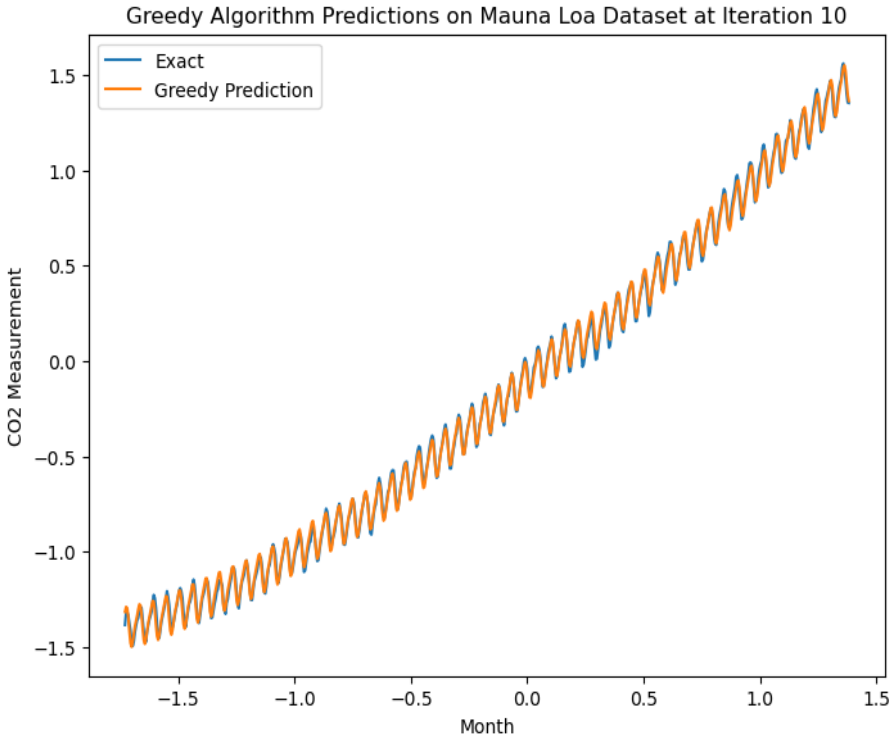
This section of the appendix presents the improvements to the greedy algorithm regression model as the iteration number, k, increases.

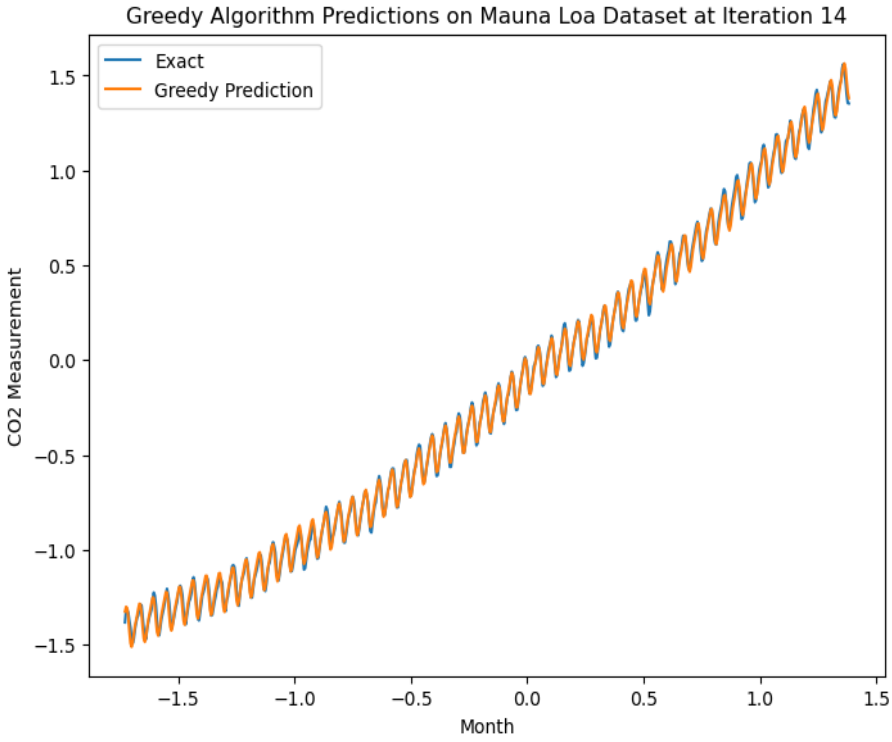



Greedy Algorithm Predictions on Mauna Loa Dataset at Iteration 2
Exact
Greedy Prediction
CO2 Measurement
Month
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5


Greedy Algorithm Predictions on Mauna Loa Dataset at Iteration 5
Exact
Greedy Prediction
CO2 Measurement
Month
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5


Greedy Algorithm Predictions on Mauna Loa Dataset at Iteration 8
Exact
Greedy Prediction
CO2 Measurement
Month
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5


Greedy Algorithm Predictions on Mauna Loa Dataset at Iteration 10
Exact
Greedy Prediction
CO2 Measurement
Month
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5


Greedy Algorithm Predictions on Mauna Loa Dataset at Iteration 14
Exact
Greedy Prediction
CO2 Measurement
Month
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5